\section{RYU SDN Controller}

SDN controller is the brain of the SDN ecosystem with all the network-wide control functions aggregated at the controller as a global snapshot which is then presented as a single logical switch to the application domain. There have been several implementations of the controller since the inception of the SDN. Some of the salient features of an SDN controller include the ability to scale out the network without bounds, the ability to support network programming in a vendor agnostic manner, protocol agnostic characteristic that allows the design and use of new southbound APIs and so on. Some of the open-sourced, adopted controllers are OpenDayLight, ONOS, Floodlight, RYU. As part of the thesis work, RYU is chosen as the controller for programming the Open vSwitch. RYU follows a simple modular design wherein applications are built and deployed as single threaded Python processes leveraging RYU's event based mechanism to interact with the other SDN applications. RYU also supports OpenFlow protocol along with various other southbound APIs which renders it just sufficient to extend the protocols, develop and deploy the desired controller application for this thesis work with a shorter development life cycle.\\\\RYU SDN controller at its core contains a set of components with well-defined APIs. The base component is the app\manager which takes up the responsibilities such as loading RYU applications, providing contexts to RYU applications and routing messages between RYU applications. It has a dedicated OpenFlow controller component to handle protocol connections to the switches. The component also generates appropriate OpenFlow events to be handled by RYU applications. The other critical component of the controller is the RYU OpenFlow wire protocol encoder and decoder \cite{RYU-Documentation}.\\\\Each RYU application is equipped with its own FIFO message queue to receive events that are processed in the order of arrival. Although the application itself is expected to be developed as a single threaded module, RYU internally spawns a thread to handle events per application module. RYU provides another way to communicate with other applications/components by exchanging contexts which are essentially application specific python objects. Since the controller is built around event-based components, RYU applications naturally are equipped to observe and generate events. An application can observe/listen to events by using a specific python decorator exposed for event handling.\\\\ As part of the work, the OpenFlow protocol's match-action capability is extended to support event types and attributes. Through this mechanism, SDN controller facilitates the switches to detect events based on the event rules. The RYU controller like its counterparts also can provide Restful API support to network applications. The application developed hence exposes the Restful APIs to commission event-based rules on Open vSwitch.